

Πανεπιστήμιο Εθνικό και Καποδιστριακό Αθηνών

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Φοιτητές:

Κυριάκος Λάμπρος Κιουράνας 1115201900238

Σοφία Δέσποινα Ζήκου 1115201900276

Περιεχόμενα

- 1. Περιγραφή Αρχείων Κώδικα και Λειτουργικότητας**
 - 3.1. Κεντρικά Modules
 - 3.2. Εισαγωγή και Εξαγωγή Δεδομένων (JSON)
 - 3.3. Στρατηγικές Steiner και Κριτήρια Εισαγωγής Σημείων
 - 3.4. Χρήσιμα Εργαλεία (Utilities)
- 2. Αναλυτική Περιγραφή Στρατηγικών Steiner**
 - 4.1. Τυπική Εισαγωγή Steiner Σημείων
 - 4.2. Περιγραφή Εφαρμοζόμενων Στρατηγικών
 - 4.3. Συγκριτική Αποτίμηση Στρατηγικών
- 3. Αλγόριθμοι Τριγωνοποίησης και Διαδικασία Βελτιστοποίησης**
 - 5.1. Delaunay Triangulation
 - 5.2. Incremental Triangulation
 - 5.3. Συζήτηση Επίδοσης: Χρόνος και Ποιότητα Πλέγματος
 - 5.4. Συνδυασμός Τριγωνοποίησης και Στρατηγικών Steiner
- 4. Υλοποίηση, Μεταγλώττιση και Εκτέλεση**
 - 6.1. Οδηγίες Εκτέλεσης
 - 6.2. Ανάπτυξη σε Περιβάλλον Παραμετροποίησης
- 5. Αποτελέσματα και Οπτικοποιήσεις**
 - 7.1. Αξιολόγηση Ποιότητας Τριγώνων
 - 7.2. Γωνίες (Οξείες, Αμβλείες) και Ενέργεια Πλέγματος
 - 7.3. Παραδείγματα Γραφημάτων
 - 7.4. Σχολιασμός Επίδοσης Ανά Στρατηγική
- 6. Διαγράμματα**

1. Περιγραφή Αρχείων Κώδικα και Λειτουργικότητας

1.1 Κεντρικά Modules

1.1.1 cgal_definitions.h

Το αρχείο αυτό συγκεντρώνει ορισμούς και typedefs που χρησιμοποιούνται σε όλο το έργο:

- **Kernel:** Ορισμός του γεωμετρικού kernel (π.χ. `Exact_predicates_inexact_constructions_kernel`).
- **Point:** Χρήση του `CGAL::Point_2<Kernel>` για τη δήλωση σημείων.
- **Polygon_2:** Ορισμός για την επεξεργασία πολυγώνων (π.χ. όταν θέλουμε να ορίσουμε πεδία ή constraint περιγράμματα).
- **CDT:** Συμβολικό όνομα για την `CGAL::Constrained_Delaunay_triangulation_2<...>` κλάση που αποτελεί τη βασική δομή για την τριγωνοποίηση.

Με αυτόν τον τρόπο, οποιαδήποτε αλλαγή στον kernel ή την προσέγγιση της ακριβούς αριθμητικής επικεντρώνεται σε ένα σημείο, διευκολύνοντας την επεκτασιμότητα.

1.1.2 CustomConstrainedDelaunayTriangulation_2.h

Πρόκειται για μια επεκτάσιμη υλοποίηση της κλάσης `Constrained_Delaunay_triangulation_2`. Ενδεικτικά:

- **Εισαγωγή σημείων χωρίς αναδιάταξη (flips):** Επιτρέπει την εισαγωγή σημείων με συγκεκριμένη σειρά και συνθήκες, χρήσιμο όταν θέλουμε να εφαρμόσουμε ειδικούς κανόνες.
- **Εφαρμογή στρατηγικών εισαγωγής σημείων:** Περιλαμβάνει hooks (ή callbacks) που ενεργοποιούνται κατά την εισαγωγή νέων σημείων, επιτρέποντας στον προγραμματιστή να επιλέξει συγκεκριμένο αλγόριθμο για την τοποθέτηση Steiner σημείων ή την ενημέρωση του πλέγματος.

1.1.3 SimpleTriangulationSearch.h

Υλοποιεί βασικές μεθόδους αναζήτησης στοιχείων σε μια τριγωνοποίηση:

- **Έλεγχος για οξείες και αμβλείες γωνίες:** Προσφέρει μια εύχρηστη συνάρτηση που υπολογίζει τις γωνίες εντός κάθε τριγώνου.
- **Επιλογή σημείων για εισαγωγή Steiner Points:** Περιλαμβάνει κανόνες (π.χ. αν ένα τρίγωνο ξεπερνά ένα όριο γωνίας, τότε σημαίνεται για βελτίωση).
- **Βελτιστοποίηση Παραμέτρων:** Μπορούν να οριστούν ανεκτές τιμές για μέγιστη αμβλεία γωνία, επιθυμητό αριθμό τριγώνων κ.λπ.

1.2 Εισαγωγή και Εξαγωγή Δεδομένων (JSON)

1.2.1 JsonLoader.h

- **Λειτουργία:** Διαβάζει δεδομένα από ένα αρχείο JSON, στο οποίο περιλαμβάνονται συνήθως:
 - Συντεταγμένες σημείων (x,y,x,y μορφή).

- Ειδικοί περιορισμοί (π.χ. λίστα από ζεύγη σημείων που ορίζουν constraint ακμές).
- Υπερπαράμετροι: όρια γωνιών, όρια ενέργειας πλέγματος, τρόπος εισαγωγής Steiner κ.ά.
- **Δυναμική φόρτωση:** Επιτρέπει στο χρήστη να αλλάζει τη διαμόρφωση της τριγωνοποίησης, χωρίς να χρειάζεται εκ νέου μεταγλώττιση.

1.2.2 JsonExporter.cpp

- **Λειτουργία:** Μετατρέπει το τελικό αποτέλεσμα (π.χ. σύνολο σημείων, ακμών και τριγώνων) σε δομή JSON, για αποθήκευση ή περαιτέρω επεξεργασία σε εξωτερικά εργαλεία.
- **Διαχείριση σφαλμάτων:** Παρέχει μηχανισμό εντοπισμού τυχόν λαθών (π.χ. αν μια ακμή αναφέρεται σε μη έγκυρο σημείο).

1.3 Στρατηγικές Steiner και Κριτήρια Εισαγωγής Σημείων

1.3.1 steiner_strategies.h

Συνοψίζει τις διαθέσιμες στρατηγικές:

1. **PERICENTER:** Υπολογισμός του περιγεγραμμένου κέντρου ενός τριγώνου (circumcenter).
2. **MAX_EDGE:** Εύρεση της μακρύτερης ακμής και τοποθέτηση σημείου στο μέσο ή σε κάποιο ποσοστό της ακμής.
3. **RANDOM:** Εισαγωγή σημείων σύμφωνα με κάποια τυχαία (συνήθως Gaussian) κατανομή στην περιοχή ενδιαφέροντος.
4. **PROJECTION:** Τοποθέτηση σημείου σε κάποιο «προβλημένο» σημείο, π.χ. στο κέντρο βάρους μιας ακμής ή τριγώνου.

1.3.2 steiner_strategies.cpp

Εδώ γίνεται η πραγματική υλοποίηση:

- **Εύρεση σημείων εντός κυρτών περιβλημάτων** (convex hull checks).
- **Αφαίρεση προβληματικών σημείων:** Αν ένα προτεινόμενο Steiner σημείο προκαλεί γεωμετρικές συγκρούσεις (π.χ. δεν υπακούει σε κάποιους constraint), τότε απορρίπτεται.

1.4 Χρήσιμα Εργαλεία (Utilities)

1.4.1 utils.hpp & utils.cpp

Περιέχουν βοηθητικές συναρτήσεις:

- **Έλεγχος κυρτότητας πολυγώνου** (π.χ. για να διαπιστωθεί αν οι ακμές ενός πολυγώνου σχηματίζουν κυρτό σχήμα).
- **Υπολογισμός γωνιών:** Μετρώντας διανύσματα ακμών, βρίσκουμε αν μια γωνία είναι οξεία, ορθή ή αμβλεία.

- **Κέντρο βάρους (centroid):** Είτε τριγώνου είτε πολυγώνου.
 - **Debugging:** Εκτυπώσεις, αναφορές, στατιστικά.
-

2. Αναλυτική Περιγραφή Στρατηγικών Steiner

2.1 Τυπική Εισαγωγή Steiner Σημείων

Η εισαγωγή Steiner σημείων σε ένα πλέγμα γίνεται με σκοπό τη βελτίωση της ποιότητας των τριγώνων ή την ικανοποίηση επιπλέον γεωμετρικών κριτηρίων. Στο πλαίσιο της Delaunay τριγωνοποίησης, τα Steiner σημεία εισάγονται σε θέσεις που δεν ανήκαν στο αρχικό σύνολο σημείων:

- **Κατά μήκος ακμών** (για να σπάσουν μεγάλες ακμές σε μικρότερες).
- **Στο εσωτερικό προβληματικών τριγώνων** (για να βελτιωθούν οξείες ή αμβλείες γωνίες).
- **Σε ειδικές θέσεις** που ορίζει ο χρήστης (π.χ. τοπογραφική αναβάθμιση συγκεκριμένου τμήματος του πλέγματος).

2.2 Περιγραφή Εφαρμοζόμενων Στρατηγικών

1. **PERICENTER (Circumcenter):** Βρίσκουμε τον κύκλο που διέρχεται από τις τρεις κορυφές του τριγώνου και το κέντρο του (circumcenter). Αυτό το κέντρο συχνά δίνει εξαιρετική βελτίωση στις γωνίες, ειδικά όταν το τρίγωνο είναι μη οξυγώνιο.
2. **MAX_EDGE:** Χρησιμοποιείται όταν επιδιώκουμε ένα είδος «ομοιόμορφης» τμηματοποίησης του πλέγματος. Αν στο πλέγμα εμφανίζεται μια ακμή με μήκος μεγαλύτερο από κάποιο κατώφλι, εισάγουμε σημείο στο μέσο της, ώστε να προκύψουν μικρότερα τρίγωνα.
3. **RANDOM:** Κατάλληλο για πειραματικές ή στοχαστικές προσεγγίσεις, όπου θέλουμε να εξερευνήσουμε πολλές πιθανές θέσεις. Χρησιμοποιείται σπάνια ως κύρια στρατηγική, αλλά μπορεί να βοηθήσει σε φαινόμενα «τοπικών ελαχίστων».
4. **PROJECTION:** Υπολογίζεται μια ενδιαφέρουσα θέση (π.χ. κέντρο βάρους τριγώνου ή μέσο ακμής) και το σημείο «προβάλλεται» ώστε να υπακούει σε συγκεκριμένα constraint (π.χ. να διατηρείται εντός ενός πολυγώνου).

2.3 Συγκριτική Αποτίμηση Στρατηγικών

- **Αποδοτικότητα:** Οι πιο “γεωμετρικά καθοδηγούμενες” (π.χ. PERICENTER, PROJECTION) τείνουν να βελτιώνουν περισσότερο τις ακραίες γωνίες.
 - **Απλότητα:** Η MAX_EDGE είναι εύκολη στην υλοποίηση και κατανοητή ως ιδέα, αλλά μπορεί να απαιτήσει πολλαπλές επαναλήψεις.
 - **Στοχαστική Προσέγγιση:** Η RANDOM μπορεί να βρει «κρυφές» θέσεις βελτίωσης, αλλά η ποιότητα του αποτελέσματος δεν είναι σταθερά προβλέψιμη.
-

3. Αλγόριθμοι Τριγωνοποίησης και Διαδικασία Βελτιστοποίησης

3.1 Delaunay Triangulation

Ο αλγόριθμος Delaunay εγγυάται ότι ο *περιγεγραμμένος κύκλος* κάθε τριγώνου δεν περιέχει κανένα άλλο σημείο στο εσωτερικό του. Βασίζεται σε:

1. **Incremental Insertion:** Εισαγωγή σημείων ένα προς ένα και ενημέρωση του πλέγματος.
2. **Divide and Conquer:** Διάρθρωση του συνόλου των σημείων σε υποσύνολα και συγχώνευση (merge) των τοπικών τριγωνοποιήσεων.
3. **Flipping edges:** Τοπικές ανταλλαγές ακμών (edge flips) που βελτιώνουν τυχόν μη-Delaunay τριγωνάκια.

3.2 Incremental Triangulation

Η Incremental τριγωνοποίηση είναι πιο απλή και συχνά χρησιμοποιείται σε εφαρμογές όπου δεν είναι απαραίτητη η αυστηρή ιδιότητα της Delaunay. Απλώς εισάγονται σημείων διαδοχικά:

- Ξεκινάμε με ένα αρχικό τρίγωνο (ή πολυώνυμο).
- Προσθέτουμε ένα σημείο στο πλέγμα και διαιρούμε το τρίγωνο στο οποίο έπεσε το σημείο σε μικρότερα τμήματα.
- Δεν εκτελούνται απαραίτητα edge flips για να επιτευχθεί η ιδιότητα Delaunay.

3.3 Συζήτηση Επίδοσης: Χρόνος και Ποιότητα Πλέγματος

- **Delaunay:** Παρέχει συνήθως καλύτερη ποιότητα, αλλά συνοδεύεται από περισσότερες πράξεις flip. Οι περισσότεροι αλγόριθμοι έχουν μέση πολυπλοκότητα $O(n \log n)$.
- **Incremental:** Αρκετά γρήγορη για μικρά datasets, απλούστερη στην υλοποίηση, αλλά η ποιότητα των τριγώνων μπορεί να χρειαστεί επιπλέον παρεμβάσεις (π.χ. εισαγωγή Steiner ή flips).

3.4 Συνδυασμός Τριγωνοποίησης και Στρατηγικών Steiner

Η βασική προσέγγιση είναι η εξής:

1. **Αρχική Τριγωνοποίηση** (Delaunay ή Incremental).
2. **Αναζήτηση Προβληματικών Τριγώνων** με κριτήριο γωνιών ή επιφανειών.
3. **Εισαγωγή Steiner Σημείων** (με μία από τις στρατηγικές) στα προβληματικά τρίγωνα.
4. **Επανάληψη (loop)** έως ότου δεν υπάρχουν τρίγωνα που παραβιάζουν τα κριτήρια ή έως ότου συμπληρωθεί ένας μέγιστος αριθμός επαναλήψεων.

4. Υλοποίηση, Μεταγλώττιση και Εκτέλεση

4.1 Οδηγίες Εκτέλεσης

Παράδειγμα εκτέλεσης:

```
./main -i input.json -o output.json -algorithm Delaunay -strategy PERICENTER
```

- `-i input.json`: Το αρχείο εισόδου.
- `-o output.json`: Το αρχείο εξόδου.
- `-algorithm`: Είδος τριγωνοποίησης (π.χ. Delaunay, Incremental).
- `-strategy`: Στρατηγική Steiner (π.χ. PERICENTER, MAX_EDGE, κ.ά.).

4.2 Ανάπτυξη σε Περιβάλλον Παραμετροποίησης

Το γεγονός ότι οι παράμετροι περνούν μέσω εντολών και JSON επιτρέπει:

- Δοκιμή διαφορετικών στρατηγικών σε σύντομο χρόνο.
- Αυτοματοποιημένες μετρήσεις απόδοσης (batchbatch εκτελέσεις).
- Εύκολη ενσωμάτωση σε pipelines (π.χ. CI/CD για projects γεωπληροφορικής).

5. Αποτελέσματα και Οπτικοποιήσεις

5.1 Αξιολόγηση Ποιότητας Τριγώνων

Η ποιότητα ενός πλέγματος τριγώνων αξιολογείται συχνά βάσει:

- **Min angle / Max angle**: Πόσο κοντά στο 60° βρίσκονται οι γωνίες;
- **Aspect ratio**: Πόσο «επίμηκες» είναι τα τρίγωνα;
- **Συνολική ενέργεια πλέγματος** (εμπειρικός δείκτης ποιότητας).

5.2 Γωνίες (Οξείες, Αμβλείες) και Ενέργεια Πλέγματος

- **Οξείες γωνίες**: Μπορεί να οδηγήσουν σε αστάθεια σε αριθμητικές μεθόδους.
- **Αμβλείες γωνίες**: Δημιουργούν «αργή σύγκλιση» σε διαφορικές προσομοιώσεις και προβλήματα σε ροές (CFD).
- **Ενέργεια πλέγματος**: Ένας δείκτης που βασίζεται στο άθροισμα τετραγώνων των ακμών ή άλλων γεωμετρικών μεγεθών, έτσι ώστε να αξιολογείται πόσο «συμμετρικό» είναι το πλέγμα.

5.3 Παραδείγματα Γραφημάτων

Η εργασία συνοδεύεται από γραφήματα που απεικονίζουν τις μετρήσεις πριν και μετά την εισαγωγή Steiner σημείων. Ενδεικτικά:

- **P vs Steiner Points**: Επίδραση διαφορετικής παραμέτρου PP στον αριθμό των εισαγόμενων Steiner σημείων.

- **P vs Initial Obtuse:** Πόσες αμβλείες γωνίες παραμένουν πριν την εφαρμογή της βελτιστοποίησης.
- **Final Obtuse vs Loops:** Πώς μεταβάλλεται ο αριθμός αμβλειών γωνιών ύστερα από επαναληπτικά loops βελτιστοποίησης.
- **Energy vs Initial Obtuse:** Συσχέτιση μεταξύ της «ενέργειας» του πλέγματος και του πλήθους των αμβλειών γωνιών.

5.4 Σχολιασμός Επίδοσης Ανά Στρατηγική

- **PERICENTER:** Συνήθως έχει πολύ καλά αποτελέσματα στη μείωση των αμβλειών γωνιών, αλλά απαιτεί ακριβή υπολογισμό του κέντρου του κύκλου και διαχείριση περιπτώσεων εκτός ορίων.
- **MAX_EDGE:** Ιδανικό για «εξομάλυνση» πολύ μεγάλων ακμών, όμως αν δεν προσεχθεί, μπορεί να δημιουργήσει μεγάλο πλήθος επιπλέον σημείων.
- **RANDOM:** Αν και δεν είναι εγγυημένη η βελτίωση, σε ορισμένες περιπτώσεις μπορεί να βοηθήσει να βγούμε από «τοπικά ελάχιστα» που δεν επιλύονται εύκολα με τις deterministic στρατηγικές.
- **PROJECTION:** Εξισορροπεί την ακρίβεια (συχνά καλή ποιότητα γωνιών) με την ασφάλεια ότι δεν θα παραβιαστούν constraint.

6. Διαγράμματα

Παρακάτω παραθέτουμε τα διαγράμματα που δημιουργήσαμε με το Jupyter για κάθε σειρά αποτελεσμάτων.







































































